AMENDMENTS IN THE CLAIMS

1. (currently amended) In a data processing system having multiple processors each having a processor core, store queue (STQ) mechanism, RC mechanism, and associated processor cache, a method for facilitating cache line updates responsive to processor-issued store operations, said method comprising:

determining when a store queue entry selected for dispatch by an RC machine provides an update to an entire cache line within the processor cache, wherein said determining includes checking a value of each of a plurality of byte-enable bits associated with cache line granules that comprise the store queue entry to determine when the value of all of said byte enable bits indicate an update has occurred to a corresponding cache line granule; and

completing said update to said entire cache line of the processor cache with address-only operations, wherein no data tenure is requested on a system bus when an entire cache line is being overwritten.

2. (currently amended) In a data processing system having multiple processors each having a processor core, store queue (STQ) mechanism, RC mechanism, and associated processor cache, a method for facilitating cache line updates responsive to processor-issued store operations, said method comprising:

determining when a store queue entry selected for dispatch by an RC machine provides an update to an entire cache line by~~The method of Claim 1, wherein said determining step comprises~~:

tracking processor issued store operations gathered to said store queue entry via byte enable bits, wherein each storage granule within said store queue entry is provided a corresponding byte enable bit that is set when the storage granule is updated by a store operation;

logically ANDing each of the byte enable bits of the store queue entry to ~~determining~~ determine when all storage granules of said store queue entry have been updated; and

providing a full signal to [[the]] a STQ controller when said logically ANDing step indicates all storage granules of said entry have been updated; and

completing said update to said entire cache line with address-only operations, wherein no data tenure is requested when an entire cache line is being overwritten.

3. (original) The method of Claim 2, wherein said store queue mechanism includes [[a]] the STQ controller, said method further comprising:

tagging said entry as eligible for dispatch when said full signal is received at said STQ controller; and

enabling selection of said entry by arbitration logic of said STQ controller.

4. (original) The method of Claim 3, further comprising:

notifying the RC mechanism when said entry has been selected for dispatch; and

signaling said RC mechanism when said entry is full.

5. (original) The method of Claim 4, further comprising:

automatically resetting said byte enable bits for the entry when the entry is dispatched to an RC machine of said RC mechanism, wherein a full signal is reset to no longer indicate the entry is full.

6. (original) The method of Claim 4, wherein said completing step comprises:

receiving at said RC mechanism an entry select identifying the entry for dispatch;

receiving at said RC mechanism a signal indicating that the entry is full;

assigning the cache line update operation to an RC machine of said RC mechanism;

providing an indication to the RC machine that the entire cache line is being updated.

7. (original) The method of Claim 6, further comprising:

responsive to a receipt by said RC machine of the signal indicating that the entry being dispatched is full, activating a cache update mechanism that enables the completion of the cache line update without requiring a copy of the cache line or current data within the cache line, wherein cache line updates from entries that are not full are completed with current copies of the cache line within the cache and write permission to the cache line.

8. (original) The method of Claim 1, wherein following a miss at said processor cache or a hit at said processor cache with a cache line that becomes stale prior to completion of said update, said completing step comprises:

issuing an address-only operation to obtain write permission for said cache line; and

automatically updating said cache line with data from said entry once said write permission is obtained.

9. (original) A processor chip for utilization within a data processing system having a memory hierarchy, said processor chip comprising:

a processor core;

a store queue having multiple entries, each entry including registers for storing address and data of store operations issued by the processor core and byte-enable bits, one for each smallest storage granule of data that may be stored by a store operation;

a store queue (STQ) controller that monitors and controls said store queue;

arbitration logic associated with said STQ controller that selects an entry from among multiple eligible entries available for dispatch to be stored in a lower level cache; and

an RC mechanism that perform updates to cache lines within said lower level cache utilizing data from the entry selected for dispatch; and

first logic for determining when all storage granules within a store queue entry have received data from said processor core before said entry is selected for dispatch, wherein said first logic comprises AND logic associated with each of said entries that receives as input a value of each of said byte-enable bits and provides a single AND output that indicates when all said byte-enable bits are set, indicating a full entry; and

second logic within an RC machine of said RC mechanism assigned to update a target cache line with data of said entry for completing said update of the target cache line without initiating a data tenure on the system bus, wherein said update is completed regardless of whether said cache line is present in said lower level cache or said cache line data is stale.

10. Canceled

11. (original) The processor chip of Claim 9, said first logic comprising:

means for logically ANDing a value of each bit within said byte enabled register corresponding to said target cache line to generate an AND output;

means for providing said AND output to said STQ controller;

means, when said AND output indicates all bits have been set, for said STQ queue controller to mark said entry as eligible for dispatch.

12. (original) The processor chip of Claim 11, wherein when said entry is full and is selected for dispatch, said STQ controller further comprises:

means for signaling said RC mechanism that said entry contains a full set of data to update an entire target cache line; and

means for resetting said byte-enable bits of said entry to track a new gather of store operations.

13. (original) The processor chip of Claim 9, wherein said second logic comprises:

means for determining whether the targeted cache line is within said processor cache;

means, when said targeted cache line is within said processor cache, for issuing an address only operation to a system bus requesting write permission for said cache line; and

when said targeted cache line is not within said processor cache and said entire cache line is being updated:

means for issuing an address-only operation on said system bus to obtain write permission and invalidate all other copies of said cache line within other processor caches; and

means for writing said address and data to a line within said processor cache.

14. (original) The processor chip of Claim 9, further comprising means for tagging said targeted cache line with a most recently modified coherency state following said update.

15. (original) The processor chip of Claim 9, wherein when said entry is not a full entry, said RC machine further comprises:

means for obtaining a copy of said cache line data when said cache line data is not present within said processor cache; and

means for obtaining write permission for said cache line when said write permission is not currently owned by said processor cache, wherein said update is completed only when said cache line data is present in said cache and write permission for the cache line is obtained.

16. (currently amended) A data processing system comprising:

a processor chip having a processor core, store queue (STQ) mechanism, RC mechanism, and associated processor cache;

a memory hierarchy coupled to said processor chip and providing coherent memory operation;

means for determining when all storage granules of a cache line of the processor cache have been updated with new data from processor-issued stores, said means including means for checking a value of each of a plurality of byte-enable bits associated with the storage granules that comprise a store queue entry to determine when the value of all of said byte enable bits indicate an update has occurred to each corresponding cache line granule;

means for completing the update[[s]] to [[a]] the entire cache line of the processor cache with the data from the processor-issued stores via a single address-only system bus tenure, with no data tenure, wherein when all storage granules of said cache line are being updated by a single RC machine tenure, the update is completed without ~~requiring~~ initiating a data tenure on the system bus, regardless of whether the cache line being updated is present in the processor cache or whether the cache line is present but contains stale data.

17. (currently amended) The data processing system of Claim 16, wherein:

said store queue mechanism comprises a store queue that includes a plurality of entries, each entry having at least a data and address register and byte-enable bits; and

said processor chip further includes first logic for determining when a store queue entry selected for dispatch by an RC mechanism provides a complete update to a target cache line of the processor cache; and

said RC mechanism assigned to complete said update includes second logic for completing said update to said target cache line with address-only operations, wherein no data tenure is requested when the target entire cache line is being completely updated.

18. (currently amended) The data processing system of Claim 17, wherein said first logic for completing said determination of a full entry comprises:

~~a series of~~ AND logic associated with each of said entries that receives as input a value of each of said byte-enable bits and provides a single AND output that indicates when all said byte-enable bits are set, indicating a full entry.

19. (original) The data processing system of Claim 17, said first logic for determining comprising:

means for logically ANDing a value of each bit within said byte enabled register corresponding to said target cache line to generate an AND output; and

means for providing said AND output to a STQ controller of the store queue mechanism.

20. (original) The data processing system of Claim 19, wherein when said entry is full and is selected for dispatch, said STQ controller further comprises:

means for signaling said RC mechanism that said entry contains a full set of data to update an entire target cache line; and

means for resetting said byte-enable bits of said entry to track a new gather of store operations.

21. (original) The data processing system of Claim 17, wherein said second logic comprises:

means for determining whether the targeted cache line is within said processor cache;

means, when said targeted cache line is within said processor cache, for issuing an address only operation to a system bus requesting write permission for said cache line; and

when said targeted cache line is not within said processor cache and said entire cache line is being updated:

means for issuing an address-only operation on said system bus to obtain write permission and invalidate all other copies of said cache line within other processor caches; and

means for writing said address and data to a line within said processor cache.

22. (original) The data processing system of Claim 21, wherein when said entry is not a full entry, said RC mechanism further comprises:

means for obtaining a copy of said cache line data when said cache line data is not present within said processor cache; and

means for obtaining write permission for said cache line when said write permission is not currently owned by said processor cache, wherein said update is completed only when said cache line data is present in said cache and write permission for the cache line is obtained.